

Simulação Interativa de Fluidos Incompressíveis Utilizando Threads em CPU

Jacques Fagundes Tiago Nobrega Eros Comunello

4Vision Lab – Universidade do Vale do Itajaí (UNIVALI) – Florianópolis, SC – Brasil

{jacques.dufa,tigarmo,eros.com}@gmail.com

Abstract. *This work report performance improvements applicable to interactive fluid simulation using particles based method Smoothing Particle Hydrodynamics with the implementation of CPU (Central Processing Unit) threads to parallelize the algorithm. It also proposes the use of a method to guarantee incompressibility to evaluate its advantages in an interactive application.*

Resumo. *Este trabalho relata melhorias de performance aplicáveis a simulação de fluidos interativos utilizando o método baseado em partículas Smoothing Particle Hydrodynamics com a implementação de threads em CPU (Central Processing Unit) para a paralelização do algoritmo. Também propõe-se a utilização de um método para garantia de incompressibilidade para avaliação de suas vantagens em uma aplicação interativa.*

1. Introdução

A simulação física de fluidos sempre foi um tópico de interesse para a área de jogos digitais e animações, ter uma simulação convincente e em tempo real, além de agregar ao realismo gráfico, pode ser utilizado de diversas maneiras na mecânica de um jogo digital. Por esse motivo este artigo abordará a técnica SPH (*Smoothing Particle Hydrodynamics*), mais especificadamente tratando de líquidos, essa já bastante difundida na área de simulação de fluidos.

O SPH utiliza as equações de Navier-Stokes, a qual descreve a dinâmica de fluidos e é utilizada em duas categorias de simulação de fluidos, as Eulerianas e as Lagrangianas [Ladyzhenskaya, 1969]. Em contraste com os métodos Eulerianos que são baseados em *grids*, pontos fixos no espaço, os métodos Lagrangianos baseiam-se na interação entre partículas que acompanham o fluxo do fluido.

Na Figura 1 há um exemplo de simulação de fluidos baseada em partículas, como dito anteriormente, elas se movimentam e interagem entre si representando de maneira, dentro do possível, realística um líquido sendo agitado dentro de um recipiente.

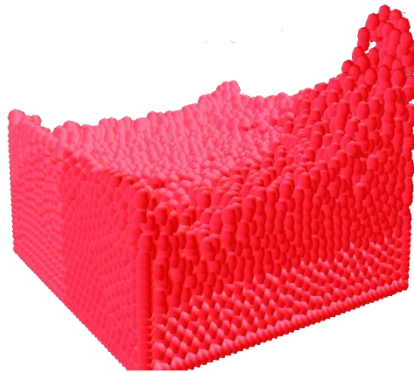


Figura 1. Simulação utilizando SPH

Apesar de o SPH ser um método já bastante estudado na literatura, diversas melhorias ainda são pesquisadas, pois por melhor que seja a técnica, a simulação é uma tarefa complexa.

Na seção 2, trabalhos relacionados ao tema deste artigo são apresentados para contexto. A seção 3 descreve as técnicas utilizadas no presente trabalho, cujos resultados de performance e comparações entre técnicas são apresentados na seção 4. Finalmente, a seção 5 discute os resultados obtidos e aponta possíveis trabalhos futuros.

2. Trabalhos Relacionados

O método SPH foi introduzido na computação gráfica após o trabalho de Stam e Fiume (1995), anteriormente era utilizado na simulação de problemas astrofísicos [Monaghan 1992]. Posteriormente foi utilizado na animação de corpos altamente deformáveis [Desbrun e Cani 1996], foi com Müller, Charypar e Gross (2003) que o método foi finalmente aplicado em uma simulação interativa de fluidos. O método proposto utilizava o SPH para a simulação de fluidos livres de superfície (líquidos), na qual deriva campos de força da equação de Navier-Stokes e modela a força de tensão na superfície do fluido exercida pelo ar da atmosfera.

Em outro trabalho foram acrescentados novos parâmetros às partículas do SPH, possibilitando a simulação de diferentes tipos e fases de fluidos interagindo entre si. Alguns resultados alcançados foram a simulação de efeitos de água fervente, ar aprisionado e dinâmica de uma lâmpada de lava. Um problema encontrado nessa abordagem foi a limitação do SPH para simular partículas individuais, que não contenham uma vizinhança [Müller *et al.* 2005].

Uma grande dificuldade no SPH é garantir de maneira eficaz a incompressibilidade de um fluido. Para isso novas abordagens que tentam adicionar essa incompressibilidade na técnica de SPH foram desenvolvidas. Uma das mais conhecidas é a PCISPH (*Predictive-Corrective Incompressible SPH*), desenvolvida por Solenthaler e Pajarola (2009), os autores criaram um método de predição e correção da pressão das partículas para melhorar a restrição de incompressibilidade de um fluido.

3. Simulação

Dentre as abordagens disponíveis para a simulação de fluidos o SPH foi escolhido, pois ao utilizar partículas ele possibilita: (i) uma fácil interação entre diferentes fluidos, por exemplo, ar e água; (ii) não restringe a movimentação do fluido a um espaço

predefinido; (iii) descarta a necessidade de garantir a conservação de massa; e (iv) descarta a resolução de um sistema de equações lineares.

3.1 SPH

O SPH, que pode ser traduzido como “hidrodinâmica de partículas suavizada”, é uma técnica para a obtenção de soluções numéricas aproximadas para as equações da dinâmica de fluidos. O método é uma substituição do fluido por um conjunto de partículas que se movem com o fluxo.

Essas partículas possuem uma distância espacial (conhecida como “distância de suavização”, geralmente representada por h), nas quais suas propriedades são atenuadas por núcleos de suavização conhecidos como *kernels* e representados por W .

Um núcleo de suavização expõe matematicamente a maneira pela qual uma partícula P e suas vizinhas dentro de um raio h ao seu redor, que pode ser referida como “vizinhança de P ”, influenciam em seu comportamento. A Figura 2 faz uma representação em duas dimensões da vizinhança em um raio h de uma partícula qualquer, quanto mais próximas as vizinhas, mais elas influenciam o seu comportamento.

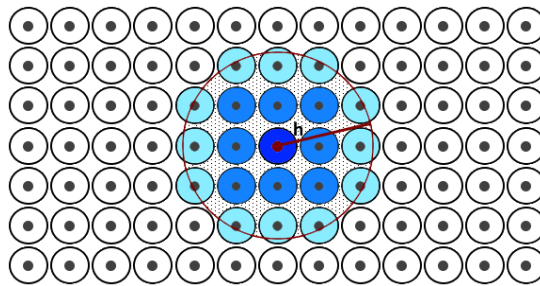


Figura 2. Vizinhança de uma partícula no SPH

O SPH consiste em discretizar as grandezas do sistema em partículas através do uso dos *kernels*, anteriormente descritos, nas partículas vizinhas. Ou seja, o estado de cada partícula depende do estado de suas partículas vizinhas. Generalizando para uma grandeza A na posição espacial r :

$$A(\mathbf{r}) = \sum_{j=1} m_j \frac{A_j}{\rho_j} W(|\mathbf{r} - \mathbf{r}_j|, h) \quad (1)$$

onde j representa uma partícula vizinha, m_j a massa da partícula vizinha, ρ_j é a densidade da partícula vizinha, W é a função de suavização utilizada, arbitrária, pois diferentes *kernels* podem ser utilizados, h é uma constante que define o raio máximo de atuação da função.

A estrutura básica dos algoritmos de SPH e os diferentes *kernels* utilizados para os cálculos são apresentados por Müller, Charypar e Gross (2003), onde a cada iteração uma sequência de passos é realizada:

- Para cada partícula, computar e armazenar uma lista de partículas vizinhas em um raio pré-determinado.

- Para cada partícula, calcular sua densidade e pressão utilizando a influência de seus vizinhos.
- Para cada partícula, calcular forças de pressão, viscosidade e forças externas (tensão de superfície, gravidade, colisões, interações).
- Para toda partícula, utilizar um método de integração para, a partir das forças previamente calculadas, encontrar a posição e velocidade resultantes.

3.2 Incompressibilidade

No método PCISPH, para garantir a incompressibilidade, é adotada uma partícula protótipo que tenha sua vizinhança completa (número ideal de partículas vizinhas), valor que pode variar entre diferentes simulações. Com ela, um valor constante de referência δ para correções é gerado previamente ao início da simulação e aplicado sempre que uma partícula não estiver dentro da variação de densidade máxima $\Delta\rho_{max}$ fornecida pelo usuário (por exemplo 5%).

Destacando que quanto maior a densidade de uma partícula, maior o valor da pressão p , e consequentemente maior a força que ela exercerá para repelir suas vizinhas. O valor de δ do PCISPH é calculado conforme:

$$\delta = \frac{-1}{\beta(-\sum_j \nabla W_{ij} \cdot \sum_j \nabla W_{ij} - \sum_j (\nabla W_{ij} \cdot \nabla W_{ij}))} \quad (2)$$

onde ∇W é o gradiente de uma função *kernel* qualquer, e β é definido por:

$$\beta = \Delta t^2 m^2 \frac{2}{\rho_0^2} \quad (3)$$

onde Δt é o *time-step* fixado no início da simulação, m é a massa de uma partícula e ρ_0 é o valor de repouso da densidade.

Para atingir essa variação de densidade determinada, uma etapa de predição é executada e repetida até a densidade ser estabilizada de acordo com a variação máxima aceita, nesse processo a pressão \hat{p} é calculada com a seguinte equação:

$$\hat{p}_i = \delta \rho_{err_i}^* \quad (4)$$

o cálculo de $\rho_{err_i}^*$ é realizado levando em consideração a diferença entre a densidade encontrada e a densidade de repouso. Finalmente todos os valores de pressão gerados na etapa de predição são somados, resultando na pressão final p da partícula:

$$p_i += \hat{p}_i \quad (5)$$

Uma grande vantagem da incompressibilidade é possibilitar que uma partícula não possua um número excessivo de vizinhos, pois quanto maior a vizinhança, maior o processamento necessário de cada iteração. Ou seja, quanto maior o número de partículas em uma simulação que utilize algum método de garantia de incompressibilidade, maior será o benefício do número reduzido de vizinhos.

3.3 Paralelização

O SPH, por ser constituído de várias partículas que são processadas em etapas imutáveis, mantendo a mesma sequencia de passos durante sua execução, é uma técnica altamente paralelizável. Copo pode ser visto na seção 4, a utilização de *threads* na simulação de fluidos utilizando SPH é recomendável, podendo melhorar em média 50% ou mais a performance geral do SPH.

Uma maneira simples e eficaz de realizar esse processo de paralelização é definir um número X de *threads* a serem utilizadas e dividir as partículas em X grupos, no qual cada *thread* seria responsável por processar um desses grupos. No entanto apesar de cada um desses grupos realizar um processamento separado, cada etapa do SPH deve ser mantida em sua ordem original por todas as partículas, sendo que nenhuma etapa pode ser iniciada sem que os todos os grupos tenham terminado a anterior. Dessa maneira é possível, caso seja necessário, adicionar diferentes propriedades a cada grupo de partículas e simular diferentes fluidos interagindo.

A Figura 3 representa um conjunto de partículas dividido em diferentes grupos e cada um dos grupos sendo processado em uma *thread* diferente para que em seguida inicie a etapa de renderização de todas as partículas juntas.

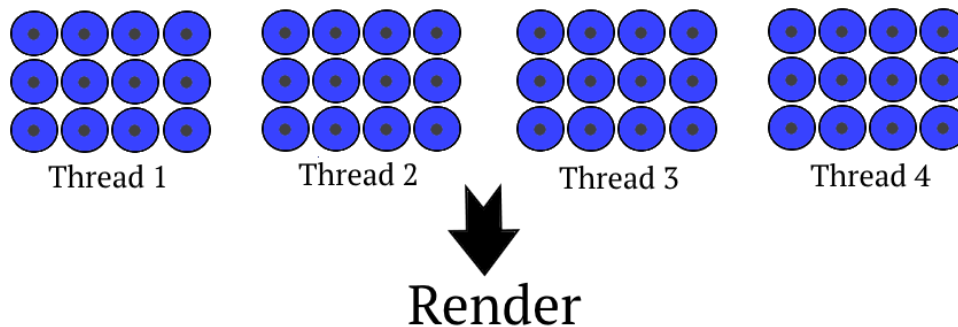


Figura 3. Grupos de partículas em etapas de processamento e renderização

4. Resultados

Os resultados a seguir ignoram os tempos de renderização e foram gerados levando em consideração apenas o processamento geral do SPH. Para os testes foi utilizado um computador com processador Intel Core I7 1.8GHz com 2 núcleos físicos e 2 virtuais.

Para todos os testes os mesmos valores de h foram utilizados, o valor de variação de densidade máxima escolhida foi de 10%, o valor de *time-step* utilizado no SPH foi 0.008 e do PCISPH 0.005. O número de *threads* a serem utilizadas foi definido em quatro, pois em comparação com outros números, esse gerou os melhores resultados levando em consideração as configurações do processador utilizado.

A Tabela 1 demonstra resultados gerados com 1.000 iterações, levando em consideração diferentes quantidades de partículas $\#P$, comparando em milissegundos (*ms*) o tempo médio de duração para uma iteração dos métodos, com e sem threads. Por fim, também são comparadas as variações de densidade ρ_{erri} de cada simulação.

Tabela 1. Resultados de várias simulações utilizando SPH e PCISPH com e sem threads.

Método	# <i>P</i>	<i>ms</i>	ρ_{erri}
SPH	1.000	2	24%
SPH - Threads	1.000	2	24%
PCISPH	1.000	4	0.2%
PCISPH - Threads	1.000	4	0.2%
SPH	10.000	87	300%
SPH - Threads	10.000	45	300%
PCISPH	10.000	85	4.6%
PCISPH - Threads	10.000	42	4.6%
SPH	20.000	307	592%
SPH - Threads	20.000	146	592%
PCISPH	20.000	204	9%
PCISPH - Threads	20.000	100	9%

Apesar dos bons resultados gerados pelo PCISPH, ele exige um maior controle do ambiente de simulação, pois caso grandes impactos sejam gerados por alguma colisão com o fluido, a etapa de predição e correção da pressão pode começar a se tornar custosa (por esse motivo a escolha do valor 10% para a variação máxima). Tal problema pode ser diluído ao diminuir o *time-step* da simulação, constante que é responsável por definir quão grande é o tempo decorrido entre cada iteração do algoritmo.

Ao comparar o SPH original com o PCISPH nota-se que se aplicados em uma aplicação interativa, a escolha da técnica dependerá dos resultados que se deseja alcançar com a simulação. Caso o fluido simulado seja em pequenas escalas, apenas para geração de alguns efeitos sem a necessidade de garantir a incompressibilidade, uma simples implementação de SPH é a melhor opção, podendo alcançar o dobro de desempenho em comparação ao outro método destacado. Porém, caso necessite de uma grande quantidade de partículas em uma simulação que requer maior realismo, uma técnica como o PCISPH é a mais indicada apesar de possuir uma maior instabilidade, pois quanto maior o número de partículas ela tende a ser mais vantajosa que o SPH comum.

5. Conclusão

A comparação proposta nesse trabalho relatou o funcionamento básico do SPH e do método de garantia de incompressibilidade PCISPH ambos utilizando threads em CPU (*Central Processing Unit*), além de demonstrar as vantagens do uso de cada em uma aplicação interativa que exija uma resposta rápida ao usuário.

Na Figura 4 é possível notar a grande diferença visual da compressão dos fluidos, gerando um distanciamento mais uniforme entre as partículas, e evitando regiões com agrupamentos desnecessários, como é o caso do SPH à esquerda da Figura 4.

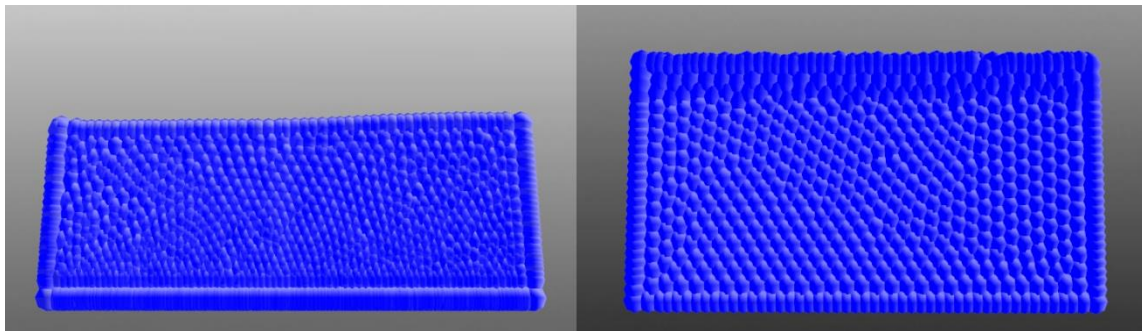


Figura 4. Demonstração do fluido simulado nos dois métodos SPH (esquerda) e PCISPH (direita).

Os resultados visuais obtidos na Figura 4 eram esperados nesse trabalho, no entanto a grande variação nos tempos de processamento do método original SPH para o PCISPH dependendo do número de partículas simulados é algo que vale ser ressaltado para futuras aplicações.

Trabalhos futuros poderiam utilizar as melhorias desenvolvidas por Ihmsen (2010) na técnica de PCISPH que sugere uma nova técnica para a colisão das partículas, controlando o problema gerado por grandes impactos na simulação e possibilitando maiores *time-steps*.

Outro ponto interessante diz respeito à comparação de outras técnicas de incompressibilidade para fluidos, destacando-se a IISPH (*Implicit incompressible SPH*) [Ihmsen *et al.* 2013] e a *Position Based Fluids* [Macklin e Müller 2013] que apresentam resultados promissores e ambas ainda foram pouco exploradas na literatura.

A utilização de simulações adaptáveis é outro fator que pode ser levado em consideração. Seja na utilização de um *time-step* adaptativo que poderia garantir uma maior estabilidade da simulação ou na utilização de partículas adaptáveis que poderiam ser processadas de maneira diferente dependendo da região que se encontram (possibilitando uma economia no tempo de processamento de cada iteração).

Vale destacar os resultados que foram alcançados por Goswami *et al.* (2010) ao utilizarem a GPU (*Graphic Processing Unit*) na implementação do SPH, conseguindo simular 75.000 partículas a 16 FPS (*Frames Per Second*). Algo possível com as recentes técnicas de paralelização conhecidas como GPGPU (*General Purpose GPU*). Portanto, uma avaliação da possibilidade de implementação do PCISPH totalmente em GPU seria interessante, podendo alcançar resultados promissores.

Referências

- Desbrun, M., Cani, M.-P. (1996). Smoothed particles: A new paradigm for animating highly deformable bodies. In Eurographics Workshop on Computer Animation and Simulation (EGCAS), R. Boulic and G. Hegron, Eds. Springer-Verlag, 61–76.
- Goswami, P., Schlegel, P., Solenthaler, B., Pajarola, R. (2010). Interactive SPH simulation and rendering on the GPU. In Proceedings of the 2010 ACM

- SIGGRAPH/Eurographics Symposium on Computer Animation, M. Otaduy and Z. Popovic, Eds., 1–10. 1, 3
- Ihmsen, M., Akinci, N., Gissler, M., Teschner, M. (2010). Boundary Handling and Adaptive Time-stepping for PCISPH, in Proc. VRIPHYS, pp. 79–88.
- Ihmsen M., Cornelis J., Solenthaler B., Horvath C., Teschner M. (2013). Implicit incompressible SPH. IEEE Transactions on Visualization and Computer Graphics 99, PrePrints, 1.
- Ladyzhenskaya, O. (1969). The Mathematical Theory of Viscous Incompressible Flows. 2 ed. New York: Gordon and Breach.
- Macklin, M., Müller, M. (2013). Position based fluids, ACM Transactions on Graphics (TOG), v.32 n.4.
- Monaghan, J. J. (1992). Smoothed particle hydrodynamics. Annual Review of Astronomy and Astrophysics 30, 1, 543–574.
- Müller, M., Charypar, D., Gross, M. (2003). Particle-based fluid simulation for interactive applications. In SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 154–159.
- Müller, M., Solenthaler, B., Keiser, R., Gross, M. (2005). Particle-based fluid-fluid interaction. In SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation. ACM Press, New York, NY, USA, 237–244.
- Solenthaler, B., Pajarola R. (2009). Predictive-corrective incompressible sph. In SIGGRAPH '09: ACM SIGGRAPH 2009 papers, ACM, pp. 1–6.
- Stam, J., Fiume, E. (1995). Depicting fire and other gaseous phenomena using diffusion processes. Computer Graphics, 29(Annual Conference Series):129–136.